QA Consulting

QA Consulting.

Hello World

IN SPRING BOOT

Prepared by
Dale Carr

☎ +447534789613

✉ dale.carr@qa.com

Contents

Getting Started

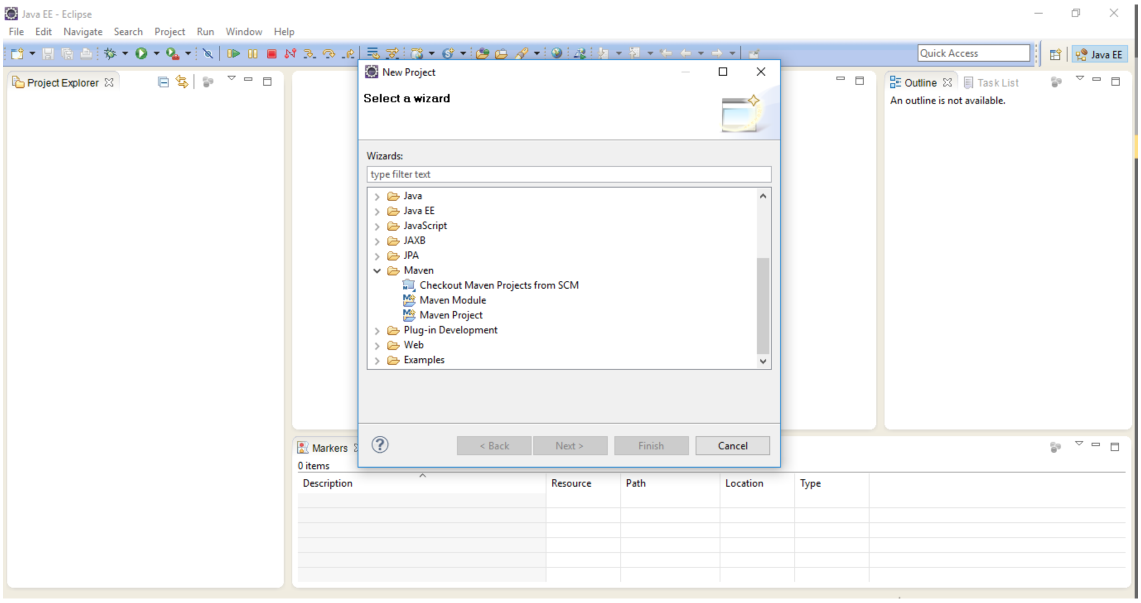
There are a few ways of building a simple app with Spring Boot

1. There are plugins/shortcuts to start a spring boot project in eclipse that will auto add some dependencies they are spring initializer –https://start.spring.io
2. STS /STS eclipse plugin
3. Manually (Which we'll be covering)

Creating a new project

IN eclipse File>New>Maven Project

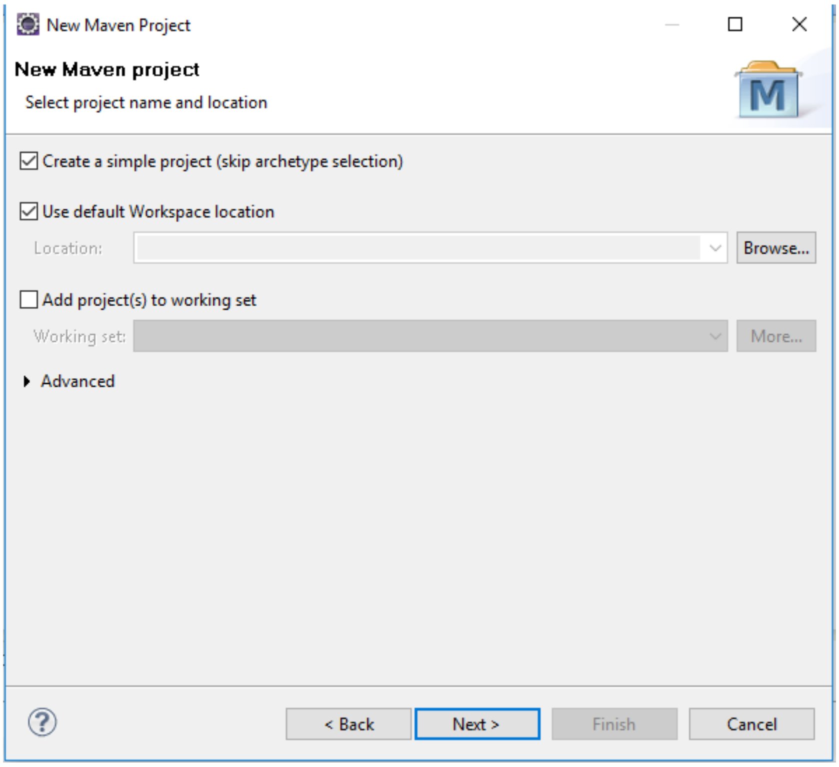
4. Select the following options:



- Create a simple project (skip archetype selection)
- Use default Workspace location

5. Leave the following options unselected

- Add project(s) to working set



New Maven Project

New Maven project

Select project name and location

☑ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location:

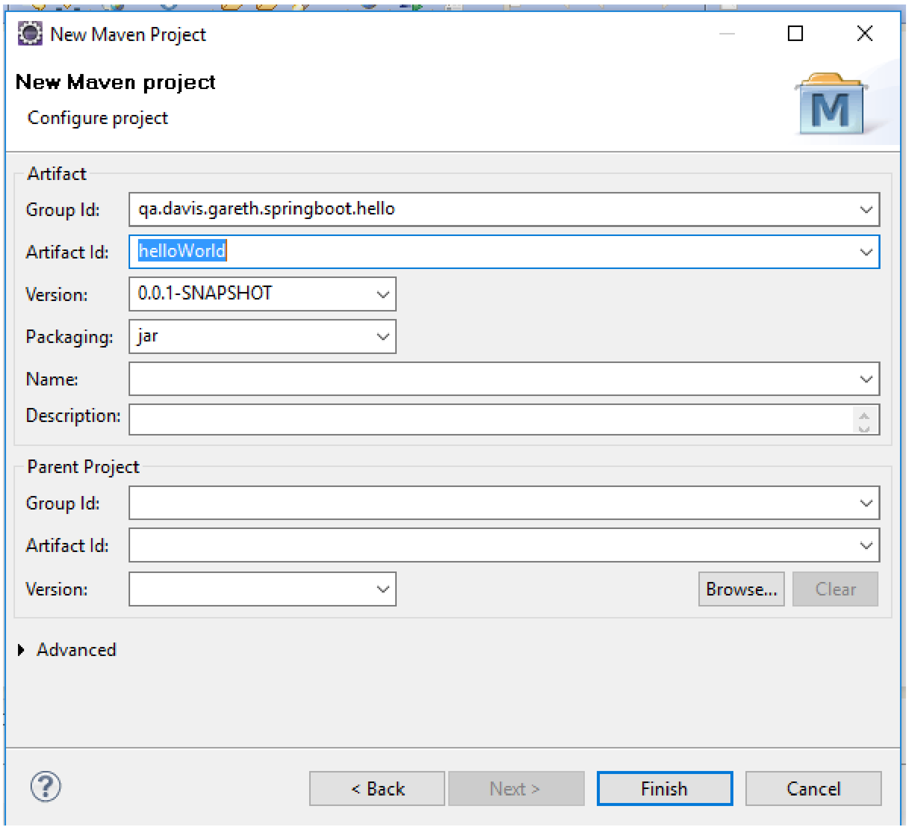
□ Add project(s) to working set

Working set:

6 Select next

7 Enter your group ID and Artifact ID (and version if you want to)

(Group ID is how you uniquely name this project across all projects. Normally this will start with your company domain in reverse going down to your team and project Artifact id is the name of the project (or Jar) without versions)



New Maven Project

New Maven project

Configure project

Artifact

Group Id: qa.davis.gareth.springboot.hello

Artifact Id: helloWorld

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Parent Project

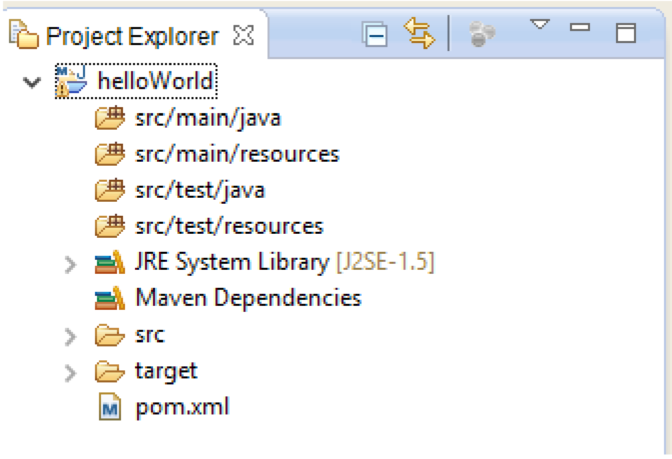
Group Id:

Artifact Id:

Version:

8 As you should be aware from the maven course this is just how you create a barebones maven project without any dependencies.

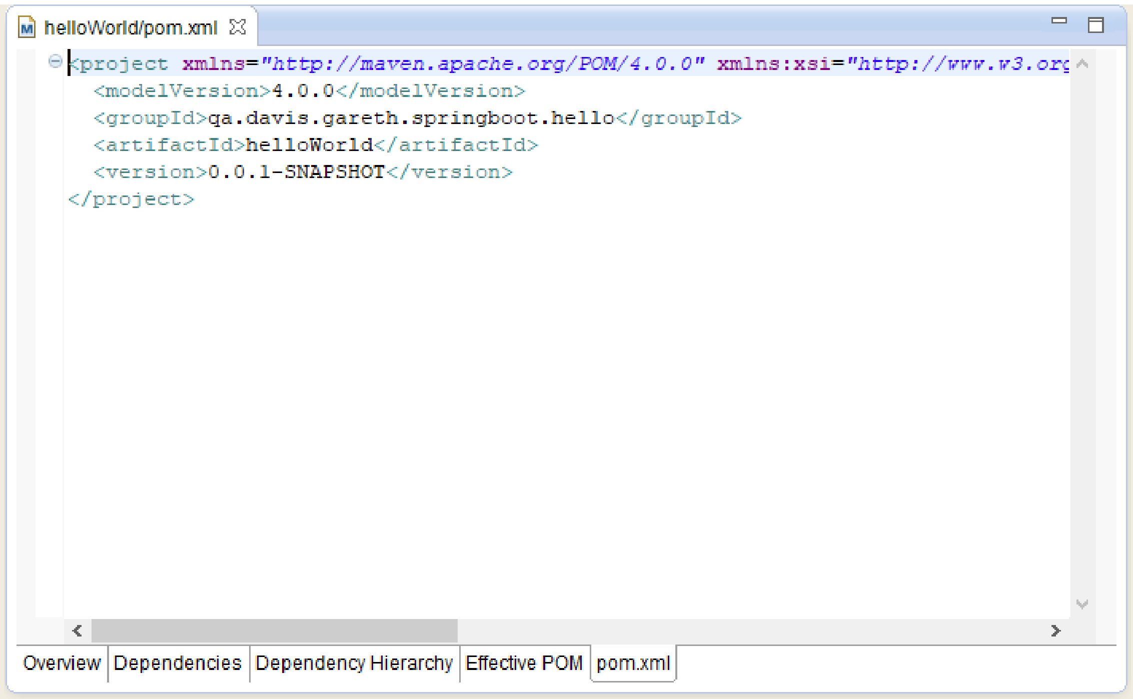
9 Your project tree structure should look like the following;



Project Explorer

helloWorld
- src/main/java
- src/main/resources
- src/test/java
- src/test/resources
- JRE System Library [J2SE-1.5]
- Maven Dependencies
- src
- target
- pom.xml

10 If you open the POM it should look like the following;

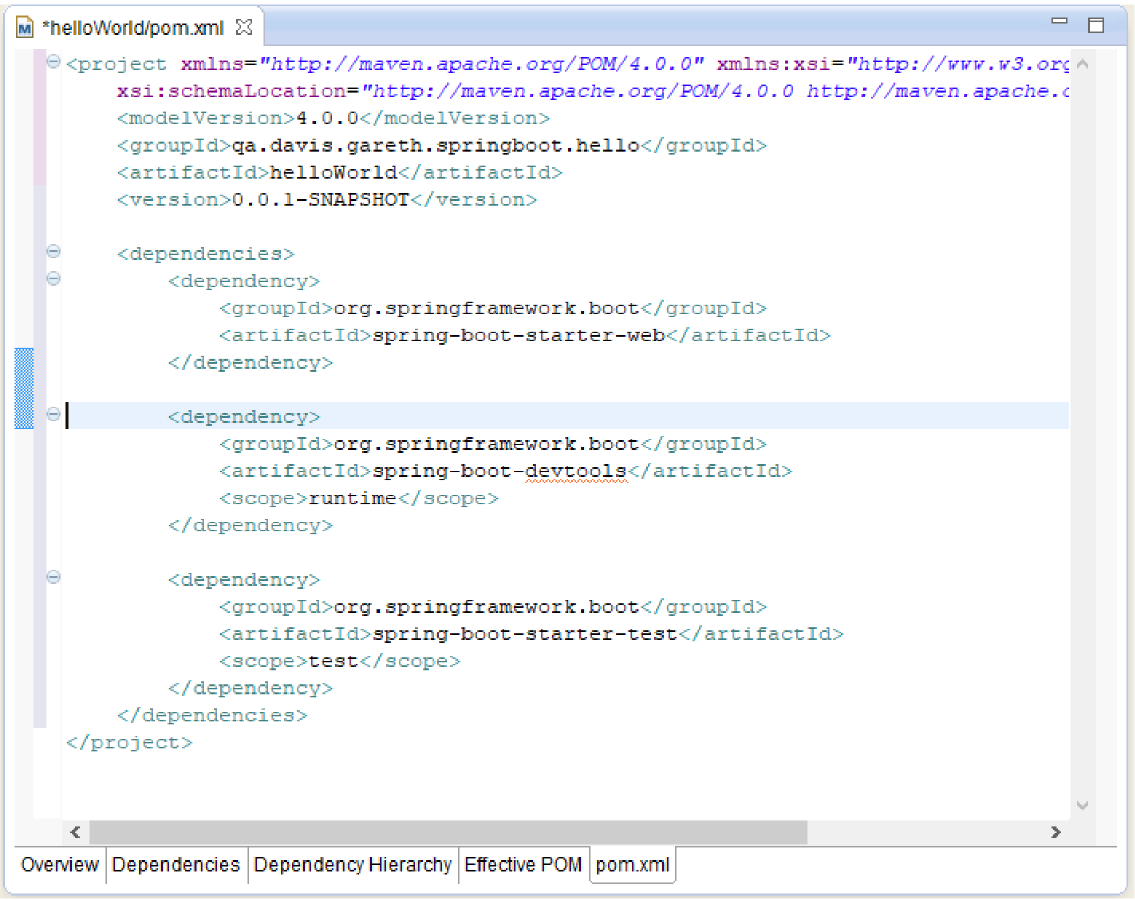


```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
  <modelVersion>4.0.0</modelVersion>
  <groupId>qa.davis.gareth.springboot.hello</groupId>
  <artifactId>helloWorld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

11 We need to add three dependencies into the pom.xml

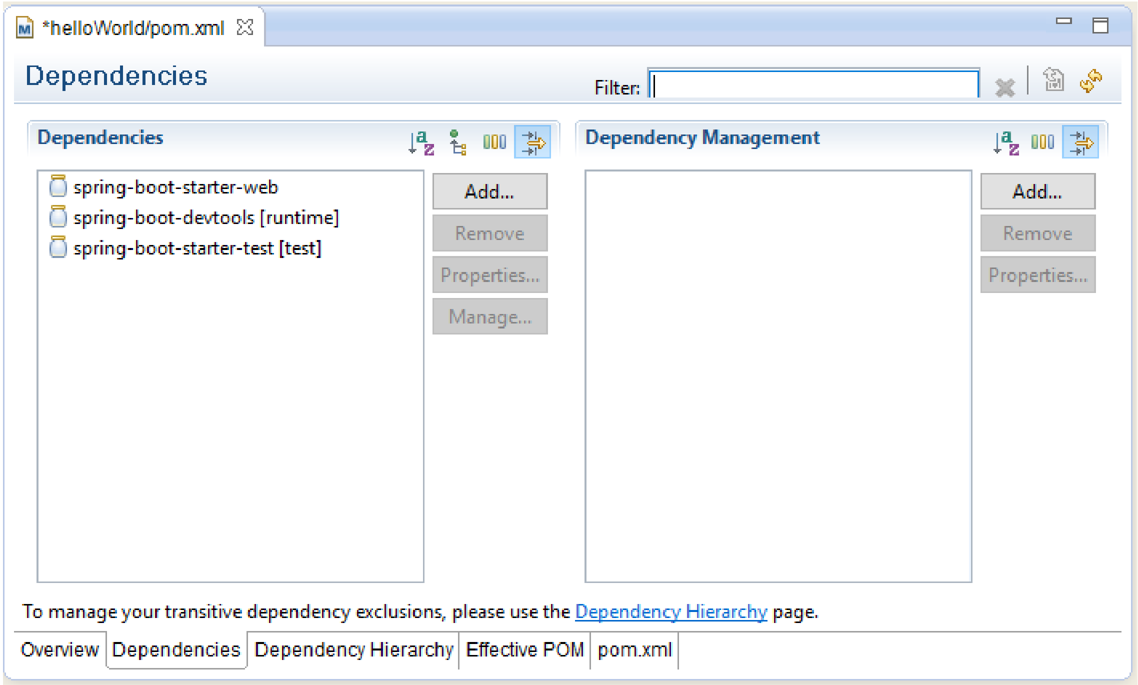
- Starter Web, Starter Test and dev Tool
- Starter web allows us to build Spring Boot web applications or REST services
- Spring boot devtools allows us to quickly set up and run services and includes functionality like live reload and global settings
- Starter test bring unit testing into the project using Spring Test, Mockito and Junit
- Add the dependencies using either the dependencies tab or by directly editing the POM.xml file

12 Until your POM.xml looks like this



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.o
    <modelVersion>4.0.0</modelVersion>
    <groupId>qa.davis.gareth.springboot.hello</groupId>
    <artifactId>helloWorld</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

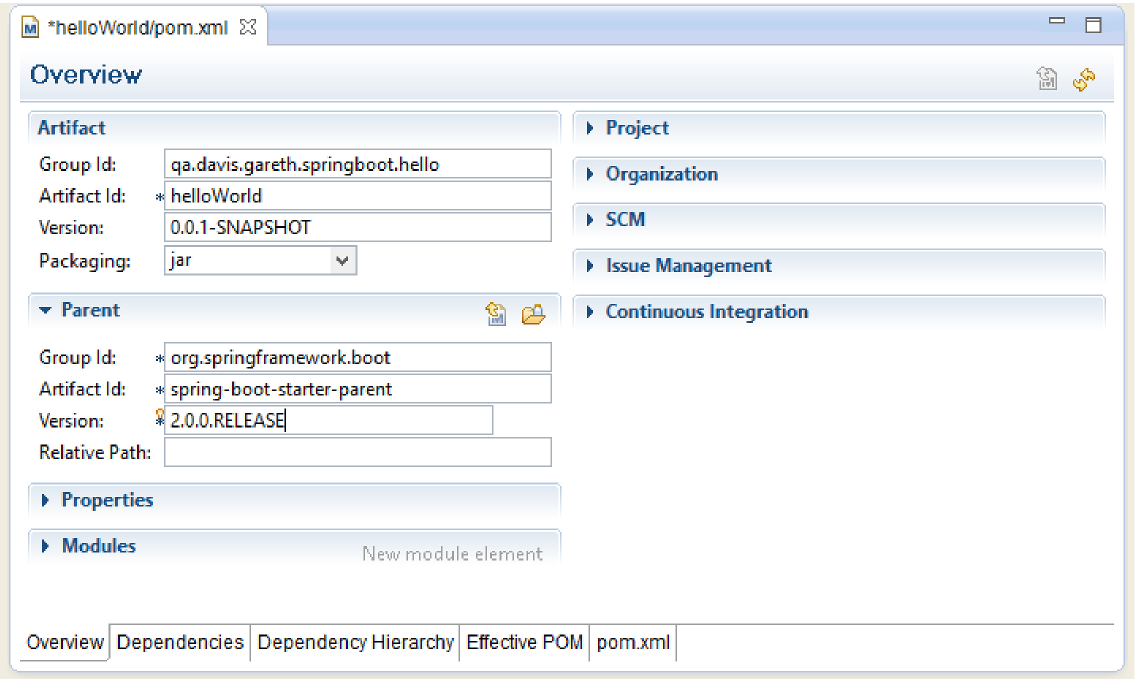
13 And your dependencies page looks like this



spring-boot-starter-web
spring-boot-devtools [runtime]
spring-boot-starter-test [test]

To manage your transitive dependency exclusions, please use the Dependency Hierarchy page.

14 We haven't added the version numbers for the dependencies so that they can be auto controlled by the Spring Boot Starter parent



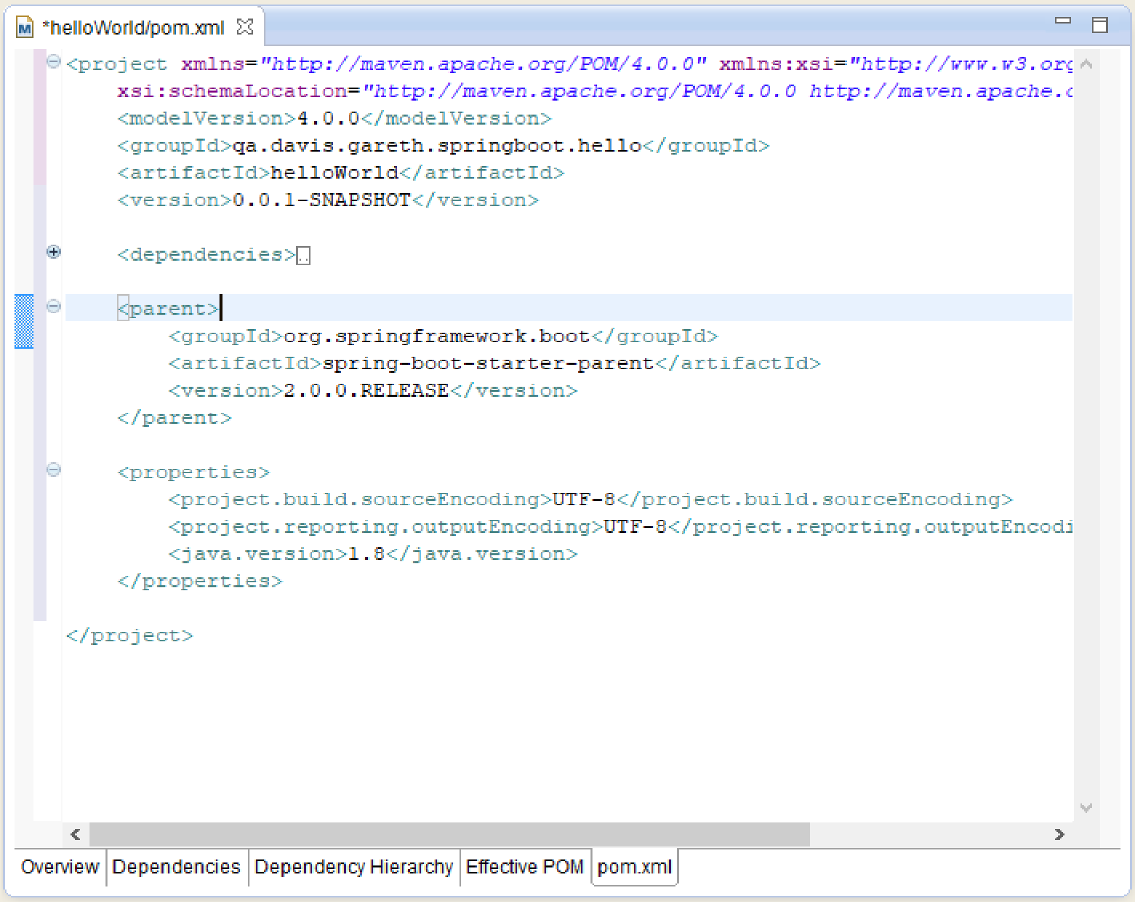
Artifact

Group Id: qa.davis.gareth.springboot.hello
Artifact Id: * helloWorld
Version: 0.0.1-SNAPSHOT
Packaging: jar

Parent

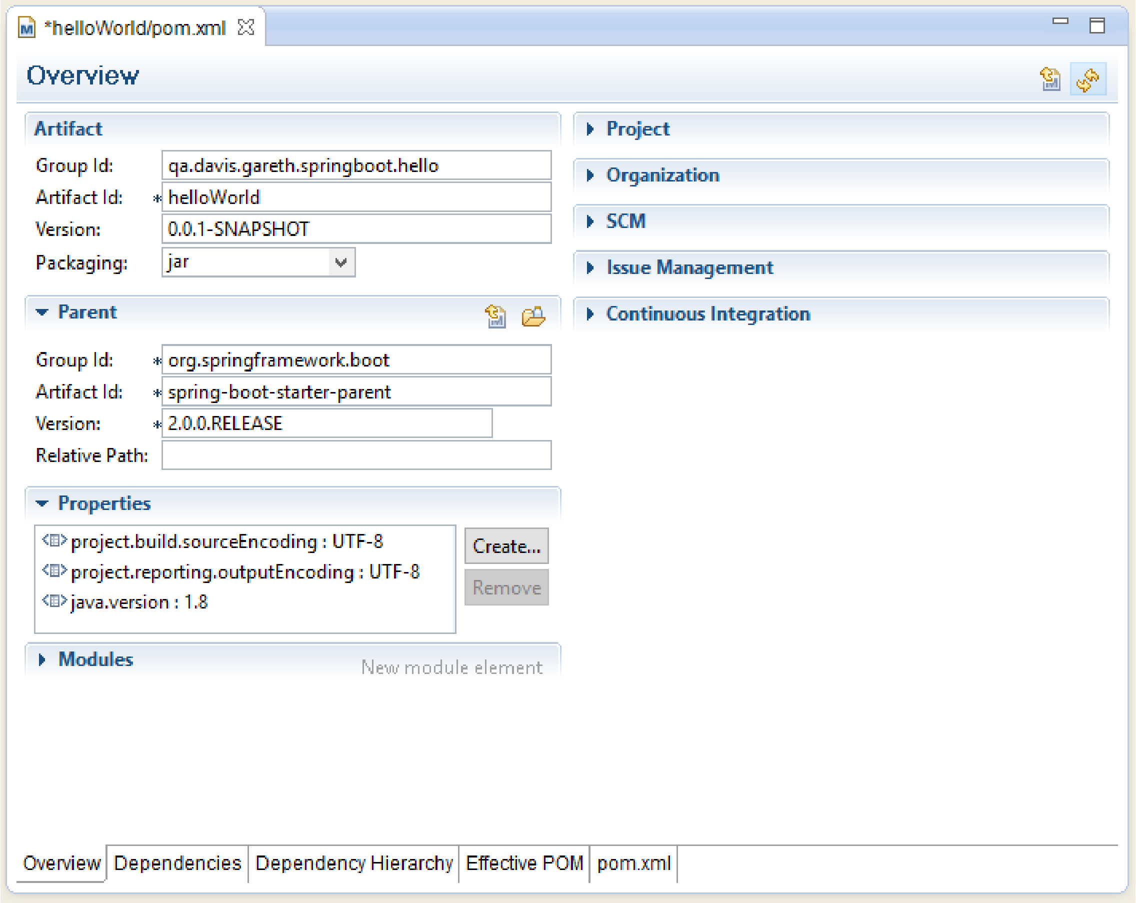
Group Id: * org.springframework.boot
Artifact Id: * spring-boot-starter-parent
Version: * 2.0.0.RELEASE
Relative Path:

15 And we want to add our Java version as well.



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.c
    <modelVersion>4.0.0</modelVersion>
    <groupId>qa.davis.gareth.springboot.hello</groupId>
    <artifactId>helloWorld</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <dependencies>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.0.0.RELEASE</version>
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncodi
        <java.version>1.8</java.version>
    </properties>

</project>
```

16 Once this is completed you should have an overview as follows



Artifact

Group Id: qa.davis.gareth.springboot.hello
Artifact Id: * helloWorld
Version: 0.0.1-SNAPSHOT
Packaging: jar

Parent

Group Id: * org.springframework.boot
Artifact Id: * spring-boot-starter-parent
Version: * 2.0.0.RELEASE
Relative Path:

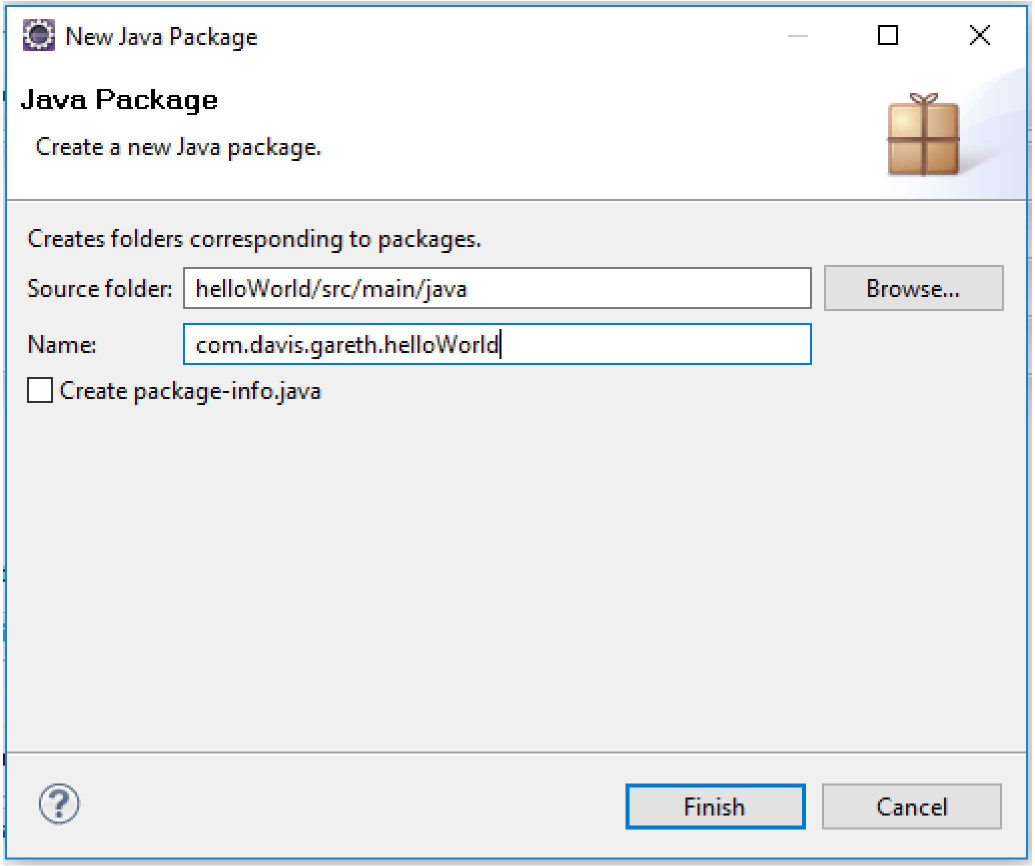
Properties

project.build.sourceEncoding : UTF-8
project.reporting.outputEncoding : UTF-8
java.version : 1.8

17 And you are finally ready to build the first app!

Creating the App

1 Create a new package



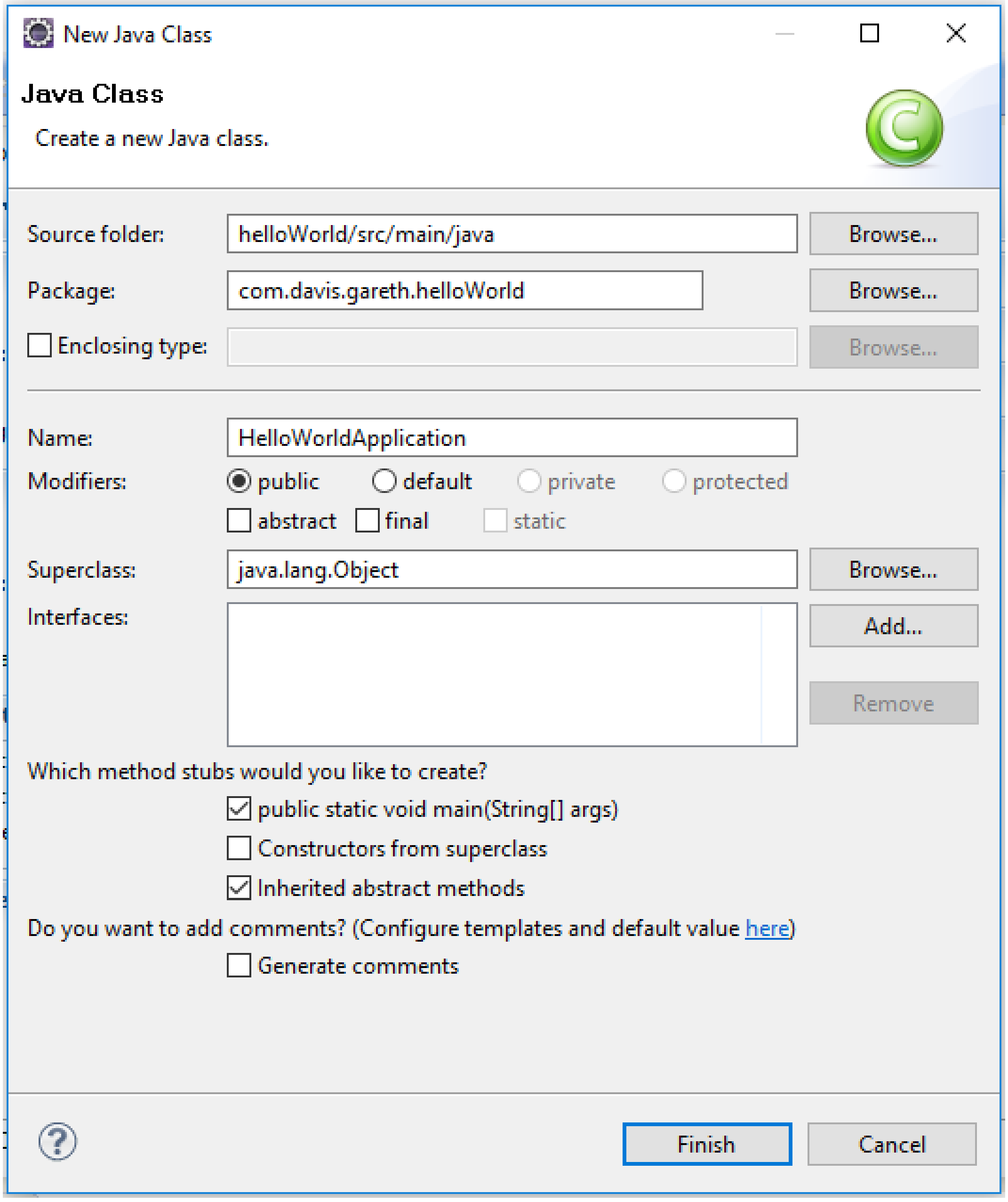
Creates folders corresponding to packages.

Source folder: helloWorld/src/main/java

Name: com.davis.gareth.helloWorld

□ Create package-info.java

2 Create an Application class



Java Class

Create a new Java class.

Source folder: helloWorld/src/main/java

Package: com.davis.gareth.helloWorld

□ Enclosing type:

Name: HelloWorldApplication

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

□ Constructors from superclass

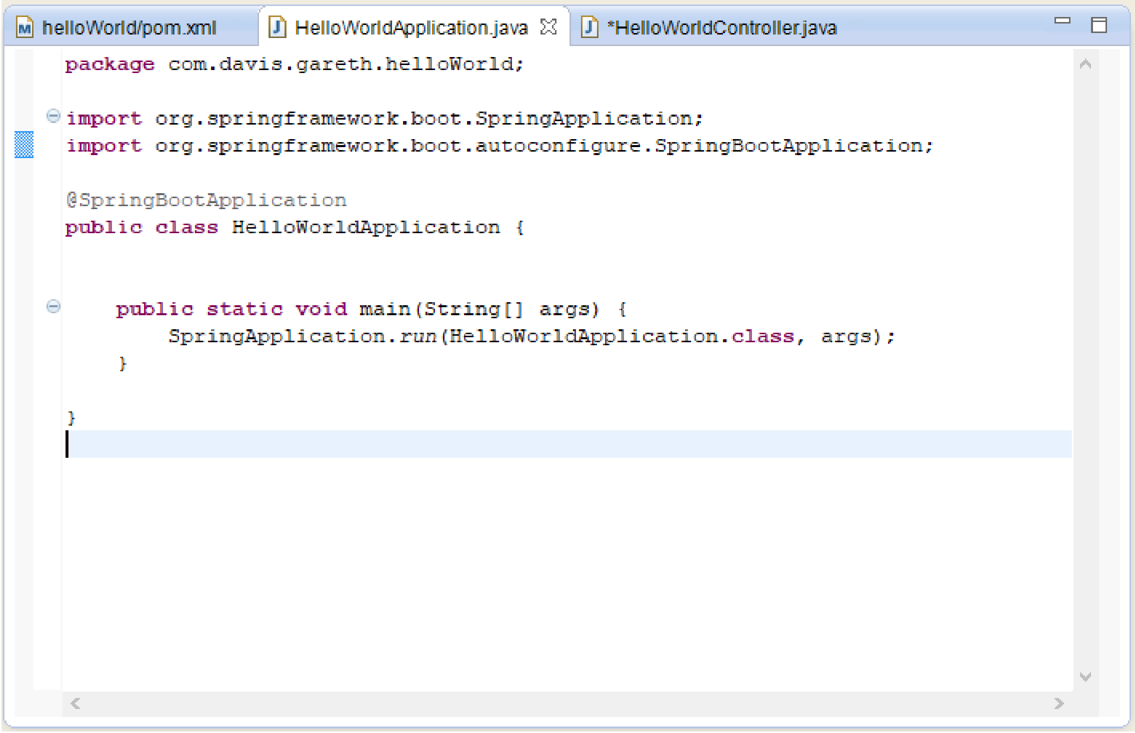
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Your application class needs a static void main as this will be the entrance point for your application.

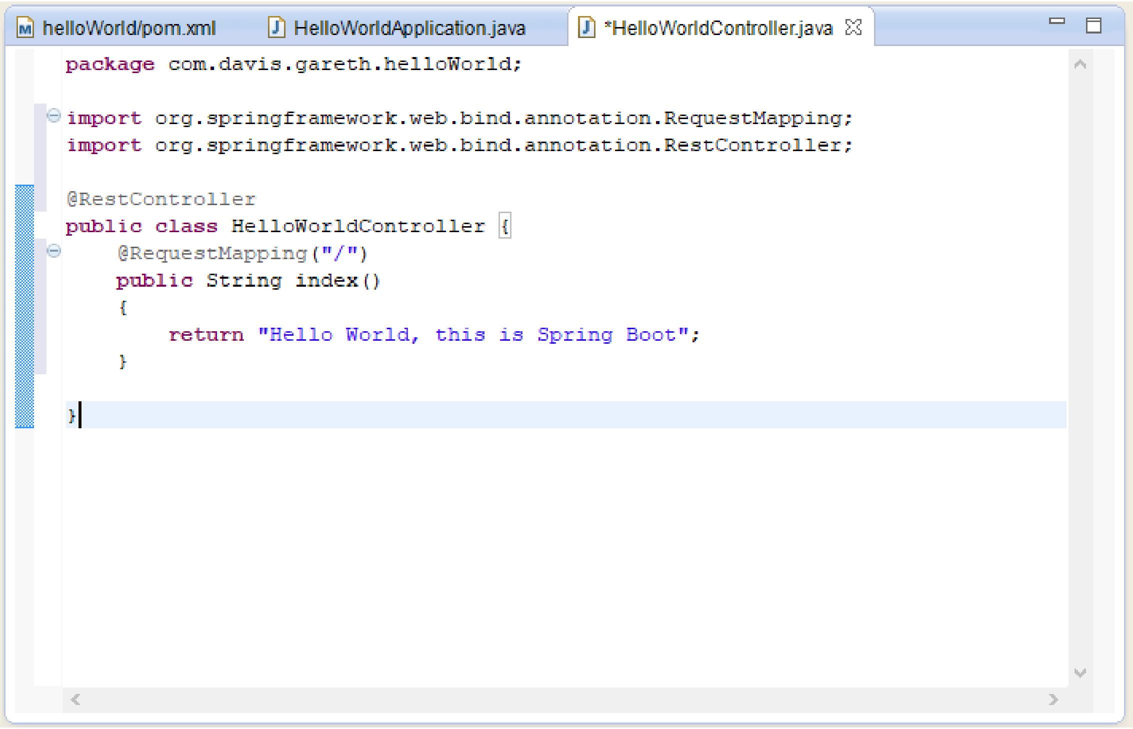
3 Add the following code into the class



```
package com.davis.gareth.helloWorld;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class HelloWorldApplication {


    public static void main(String[] args) {
        SpringApplication.run(HelloWorldApplication.class, args);
    }

}
```

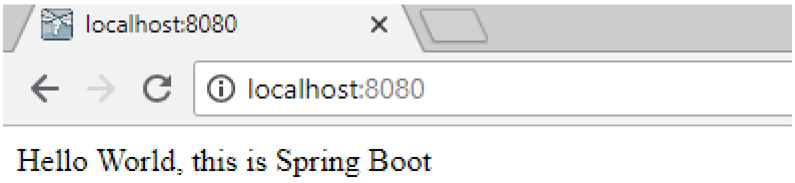
4 Create a new class called HelloWorldController.java

5 Add the following code into the class



```
package com.davis.gareth.helloWorld;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloWorldController {
    @RequestMapping("/")
    public String index()
    {
        return "Hello World, this is Spring Boot";
    }

}
```

6 Now run HelloWorldApplication.java and if you go to localhost:8080 you should see the following message:



Hello World, this is Spring Boot